

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2004-245988

(P2004-245988A)

(43) 公開日 平成16年9月2日(2004.9.2)

(51) Int.Cl.⁷
G09C 1/00F I
G09C 1/00 610Aテーマコード (参考)
5J104

審査請求 未請求 請求項の数 12 O L (全 12 頁)

(21) 出願番号 特願2003-34591 (P2003-34591)
(22) 出願日 平成15年2月13日(2003.2.13)(71) 出願人 000002185
ソニー株式会社
東京都品川区北品川6丁目7番35号
(74) 代理人 100094053
弁理士 佐藤 隆久
(72) 発明者 金丸 昌司
東京都品川区北品川6丁目7番35号 ソ
ニー株式会社内
Fターム(参考) 5J104 JA05

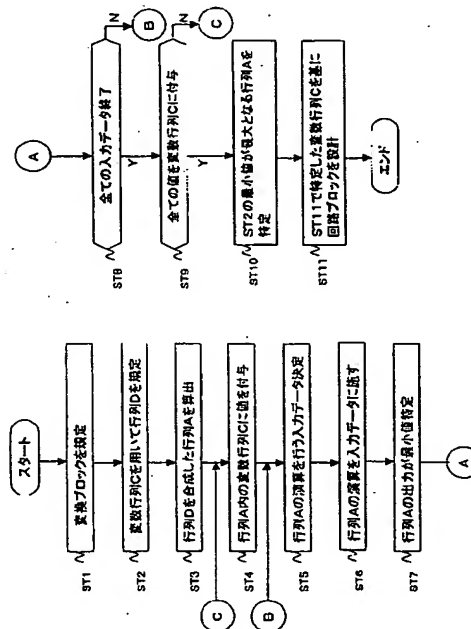
(54) 【発明の名称】 データ処理装置、その方法およびそのプログラムと線形変換回路および暗号化回路

(57) 【要約】

【課題】 アクティブS-box数を最大にする線形変換を特定するデータ処理方法を提供する。

【解決手段】 複数の線形変換候補のうち、当該線形変換候補を実現する回路上の制約を満たす複数の前記線形変換候補を特定し(ST1, ST2, ST3)、当該特定した線形変換候補のそれぞれについて複数の入力データを基に線形変換処理を行い、それらの処理結果内に生じる零の数の最小値(いわゆるアクティブS-box)を求め(ST4~ST9)、その最小値を最大とする線形変換候補を特定する(ST10)。そして、当該特定した線形変換候補を基に、線形変換部を構成する(ST11)。

【選択図】 図4



【特許請求の範囲】

【請求項 1】

線形変換を実現する回路上の制約をそれぞれ満たす複数の線形変換候補を特定する第 1 の工程と、

前記第 1 の工程で特定した前記複数の線形変換候補のそれぞれについて、複数の入力データに対して当該線形変換候補が規定する線形変換を行ったそれぞれの結果に零が生じる個数の最小値を特定する第 2 の工程と、

前記第 1 の工程で特定した前記複数の線形変換候補のうち、前記第 2 の工程で特定した前記最小値が最大となる前記線形変換候補を特定する第 3 の工程と
を有するデータ処理方法。

10

【請求項 2】

前記第 1 の工程は、前記複数の線形変換候補として、置換行列内の 2 つの零領域のうち一方を変数行列で置き換えた複数の単位線形変換の合成である線形変換を特定し、

前記第 2 の工程は、前記複数の単位線形変換の前記変数行列として異なる複数の行列を付与して得られる前記線形変換候補の各々について、前記最小値を特定する

請求項 1 に記載のデータ処理方法。

【請求項 3】

前記複数の単位線形変換の数が M であり、

前記単位線形変換は、M 行 M 列の行列演算で実現する

請求項 2 に記載のデータ処理方法。

20

【請求項 4】

前記第 1 の工程は、前記線型変換候補を、変数行列 C_1 , C_2 , C_3 , C_4 を用いて下記式 (1) で規定する

請求項 3 に記載のデータ処理方法。

【数 1】

$$\begin{pmatrix} I + C_4 C_3 + C_2 C_1 + C_4 C_3 C_2 C_1 + C_4 C_1 & C_2 + C_4 C_3 C_2 + C_4 \\ C_3 + C_3 C_2 C_1 + C_1 & I + C_3 C_2 \\ \dots & \dots \end{pmatrix} \quad \dots (1)$$

30

【請求項 5】

前記線形変換が、共通鍵ブロック暗号のラウンド関数処理内で規定された線形変換である場合に、

前記第 2 の工程は、平文データを非線形拡散処理して得られた前記入力データに対して前記線形変換を行う

請求項 1 に記載のデータ処理方法。

【請求項 6】

前記第 3 の工程で特定した前記線形変換候補に対応する前記単位線形変換を実現する回路ブロックを有する線形変換回路を構成する第 4 の工程

をさらに有する請求項 1 に記載のデータ処理方法。

40

【請求項 7】

コンピュータによって実行されるプログラムであって、

線形変換を実現する回路上の制約をそれぞれ満たす複数の線形変換候補を特定する第 1 の手順と、

前記第 1 の手順で特定した前記複数の線形変換候補のそれぞれについて、複数の入力データに対して当該線形変換候補が規定する線形変換を行ったそれぞれの結果に零が生じる個数の最小値を特定する第 2 の手順と、

前記第 1 の手順で特定した前記複数の線形変換候補のうち、前記第 2 の手順で特定した前記最小値が最大となる前記線形変換候補を特定する第 3 の手順と

50

を記述したプログラム。

【請求項 8】

前記第 1 の手順は、前記複数の線形変換候補として、置換行列内の 2 つの零領域のうち一方を変数行列で置き換えた複数の単位線形変換の合成である線形変換を特定し、
前記第 2 の手順は、前記複数の単位線形変換の前記変数行列として異なる複数の行列を付与して得られる前記線形変換候補の各々について、前記最小値を特定する
請求項 7 に記載のプログラム。

【請求項 9】

線形変換を実現する回路上の制約をそれぞれ満たす複数の線形変換候補を特定する第 1 の手段と、

前記第 1 の手段で特定した前記複数の線形変換候補のそれぞれについて、複数の入力データに対して当該線形変換候補が規定する線形変換を行ったそれぞれの結果に零が生じる個数の最小値を特定する第 2 の手段と、

前記第 1 の手段で特定した前記複数の線形変換候補のうち、前記第 2 の手段で特定した前記最小値が最大となる前記線形変換候補を特定する第 3 の手段と
を有するデータ処理装置。

【請求項 10】

共通鍵ブロック暗号のラウンド関数処理内で規定された線形変換を行う線形変換回路であって、

複数のデータのそれぞれに対応した複数のデータ線と、

前記複数のデータ線を介して入力された前記複数のデータに線形変換を順に施す複数の回路ブロックと

を有し、

前記回路ブロックの各々は、前記複数のデータ線のうち一部の複数の前記データ線上に設けられた演算回路を有し、演算回路が設けられていない各々の前記データ線から最大 1 個の前記演算回路に対してデータが供給されるように構成されている
線形変換回路。

【請求項 11】

前記回路ブロックの各々は、前記複数のデータ線を同数の第 1 のデータ線群と第 2 のデータ線群とに分けた場合に、前記第 1 のデータ線群内の前記データ線にのみ前記演算回路を有し、前記第 2 のデータ線群内の前記データ線から、前記第 1 のデータ線群内の前記演算回路にデータを出力するように構成されている

請求項 10 に記載の線形変換回路。

【請求項 12】

ラウンド関数処理を行って共通鍵ブロック暗号化を行う暗号化装置であって、

前記ラウンド関数処理内で規定された非線形変換を行う非線形変換回路と、

前記非線形変換回路により前記非線形変換が施された入力データに対して線形変換を行う線形変換回路と

を有し、

前記線形変換回路は、

前記入力データを構成する複数のデータのそれぞれに対応した複数のデータ線と、

前記データ線を介して入力された前記複数のデータに線形変換を順に施す複数の回路ブロックと

を有し、

前記回路ブロックの各々は、前記複数のデータ線のうち一部の複数の前記データ線上に設けられた演算回路を有し、演算回路が設けられていない各々の前記データ線から最大 1 個の前記演算回路に対してデータが供給されるように構成されている

暗号化装置。

【発明の詳細な説明】

【0001】

10

20

30

40

50

【発明の属する技術分野】

本発明は、暗号化処理などで規定された線形変換を行う線形変換回路の設計に用いられるデータ処理方法、装置、そのプログラムと、線形変換回路および暗号化装置に関する。

【0002】

【従来の技術】

情報セキュリティを達成するために種々の暗号化技術開発されている。

このような暗号化技術の一種である共通鍵ブロック暗号は、例えば、非線形処理と線形処理（拡散処理）とからなるラウンド関数を規定している。

上記ラウンド関数の非線形処理は、S-boxと呼ばれるユニットで構成され、入出力間の非線形性を実現している。

また、上記ラウンド関数の線形処理は、多ビットからなる入力データの影響を複数ビットに拡散させる線形変換を行う。

このような線形変換を用いる方法として、AES (Advanced Encryption Standard) 等で用いられるMDS (Maximal Distance Separable) を利用したものがある。

MDSは、 $GF(2^8)$ 等の拡大体上の変換を用いることによって、効率よくビット拡散を行う手法である。

しかしながら、MDSは、実装時に回路構成が複雑になるという欠点がある。

このような決定を解消するものとして、CamelliaおよびE2などの暗号手法がある。この暗号手法では、高速かつ小規模な構成の回路を構成するために、 $GF(2)$ 上の

【0003】

【特許文献1】

特開2002-91295号公報

【0004】

【発明が解決しようとする課題】

しかしながら、 $GF(2)$ 上の変換による高い拡散効率を得るために、いわゆるアクティブ (Active) S-box数を最大にする回路構成を、線形変換を実現する回路上の制約とは無関係に、全ての線形変換候補について演算を行って決定しており、膨大な計算量が必要になるという問題がある。

ここで、アクティブS-box数は、複数の入力データに対して上記ラウンド関数の線形処理を行い、それらの処理結果内に生じる零の数の最小値である。

【0005】

本発明はかかる事情に鑑みてなされたものであり、その目的は、複数の線形変換候補のなかから、複数の入力データに線形変換を行った結果に零が生じる個数の最小値が最大となる線形変換候補を従来に比べて少ない演算量で特定できるデータ処理方法、その装置および、そのプログラムを提供することを目的とする。

また、本発明は、上述したデータ処理方法、その装置およびそのプログラムによって設計される線形変換回路および暗号化装置を提供することを目的とする。

【0006】

【課題を解決するための手段】

上述した目的を達成するために、第1の発明のデータ処理方法は、線形変換を実現する回路上の制約をそれぞれ満たす複数の線形変換候補を特定する第1の工程と、前記第1の工程で特定した前記複数の線形変換候補のそれぞれについて、複数の入力データに対して当該線形変換候補が規定する線形変換を行ったそれぞれの結果に零が生じる個数の最小値を特定する第2の工程と、前記第1の工程で特定した前記複数の線形変換候補のうち、前記第2の工程で特定した前記最小値が最大となる前記線形変換候補を特定する第3の工程とを有する。

【0007】

第1の発明のデータ処理方法の作用は以下のようなになる。

10

20

30

40

50

先ず、第1の工程において、線形変換を実現する回路上の制約をそれぞれ満たす複数の線形変換候補を特定する。

次に、第2の工程において、前記第1の工程で特定した前記複数の線形変換候補のそれぞれについて、複数の入力データに対して当該線形変換候補が規定する線形変換を行ったそれぞれの結果に零が生じる個数の最小値を特定する。

次に、第3の工程において、前記第1の工程で特定した前記複数の線形変換候補のうち、前記第2の工程で特定した前記最小値が最大となる前記線形変換候補を特定する。

【0008】

第1の発明のデータ処理方法は、好ましくは、前記第1の工程は、前記複数の線形変換候補として、置換行列内の2つの零領域のうち一方を変数行列で置き換えた複数の単位線形変換の合成である線形変換を特定し、前記第2の工程は、前記複数の単位線形変換の前記変数行列として異なる複数の行列を付与して得られる前記線形変換候補の各々について、前記最小値を特定する。

10

【0009】

第2の発明のプログラムは、コンピュータによって実行されるプログラムであって、線形変換を実現する回路上の制約をそれぞれ満たす複数の線形変換候補を特定する第1の手順と、前記第1の手順で特定した前記複数の線形変換候補のそれぞれについて、複数の入力データに対して当該線形変換候補が規定する線形変換を行ったそれぞれの結果に零が生じる個数の最小値を特定する第2の手順と、前記第1の手順で特定した前記複数の線形変換候補のうち、前記第2の手順で特定した前記最小値が最大となる前記線形変換候補を特定する第3の手順とを記述している。

20

【0010】

第2の発明のプログラムは、コンピュータによって実行され、前述した第1の発明の各工程を実現する。

【0011】

第3の発明のデータ処理装置は、線形変換を実現する回路上の制約をそれぞれ満たす複数の線形変換候補を特定する第1の手段と、前記第1の手段で特定した前記複数の線形変換候補のそれぞれについて、複数の入力データに対して当該線形変換候補が規定する線形変換を行ったそれぞれの結果に零が生じる個数の最小値を特定する第2の手段と、前記第1の手段で特定した前記複数の線形変換候補のうち、前記第2の手段で特定した前記最小値が最大となる前記線形変換候補を特定する第3の手段とを有する。

30

【0012】

第3の発明のデータ処理装置の作用は以下になる。

先ず、第1の手段において、線形変換を実現する回路上の制約をそれぞれ満たす複数の線形変換候補を特定する。

次に、第2の手段において、前記第1の手段で特定した前記複数の線形変換候補のそれぞれについて、複数の入力データに対して当該線形変換候補が規定する線形変換を行ったそれぞれの結果に零が生じる個数の最小値を特定する。

次に、第3の手段において、前記第1の手段で特定した前記複数の線形変換候補のうち、前記第2の手段で特定した前記最小値が最大となる前記線形変換候補を特定する。

40

【0013】

第4の発明の線形変換回路は、共通鍵ブロック暗号のラウンド関数処理内で規定された線形変換を行う線形変換回路であって、複数のデータのそれぞれに対応した複数のデータ線と、前記複数のデータ線を介して入力された前記複数のデータに線形変換を順に施す複数の回路ブロックとを有し、前記回路ブロックの各々は、前記複数のデータ線のうち一部の複数の前記データ線上に設けられた演算回路を有し、演算回路が設けられていない各々の前記データ線から最大1個の前記演算回路に対してデータが供給されるように構成されている。

【0014】

第5の発明の暗号化装置は、ラウンド関数処理を行って共通鍵ブロック暗号化を行う暗号

50

化装置であって、前記ラウンド関数処理内で規定された非線形変換を行う非線形変換回路と、前記非線形変換回路により前記非線形変換が施された入力データに対して線形変換を行う線形変換回路とを有し、前記線形変換回路は、前記入力データを構成する複数のデータのそれぞれに対応した複数のデータ線と、前記データ線を介して入力された前記複数のデータに線形変換を順に施す複数の回路ブロックとを有し、前記回路ブロックの各々は、前記複数のデータ線のうち一部の複数の前記データ線上に設けられた演算回路を有し、演算回路が設けられていない各々の前記データ線から最大1個の前記演算回路に対してデータが供給されるように構成されている。

【0015】

【発明の実施の形態】

以下、本発明の実施形態に係わる回路設計方法および暗号化装置について説明する。先ず、本実施形態の回路設計方法によって設計される線形変換回路を組み込んだ暗号化装置について説明する。

図1は、本実施形態の暗号化装置1の構成図である。

暗号化装置1は第5の発明の暗号化装置に対応している。

図1に示すように、暗号化装置1は、例えば、鍵生成回路2、初期処理回路3、N個のFeistel構造モジュール4_1~4_Nおよび後処理回路5を有する。ここで、Nは2以上の整数である。

暗号化装置1は、共通鍵ブロック暗号を行う。

【0016】

鍵生成回路2は、鍵データK1、K2_1~K2_N、K3を生成し、鍵データK1を初期処理回路3に出力し、鍵データK2_1~K2_NをそれぞれFeistel構造モジュール4_1~4_Nに出力し、鍵データK3を後処理回路5に出力する。

【0017】

初期処理回路3は、入力した平文データPTに初期変換を施してデータS3を生成する。データS3は、例えば、128ビットのデータである。

初期処理回路3は、データS3の下位64ビットのデータS3aをFeistel構造モジュール4_1のF関数回路11に出力し、下位64ビットのデータS3bをFeistel構造モジュール4_1のXOR(eXclusive OR:排他的論理和)回路12に出力する。

【0018】

Feistel構造モジュール4_1~4_Nの各々は、F関数回路11およびXOR回路12を有する。

Feistel構造モジュール4_1~4_Nは、直列に接続され、同じ構成を有している。

以下、Feistel構造モジュール4_1について説明する。

F関数回路11は、初期処理回路3からのデータS3aに対して、非線形処理および線形処理(拡散処理)を施して、データS11を生成し、これをXOR回路12に出力する。

F関数回路11は、本発明のラウンド関数処理を行う。

F関数回路11の構成については、後に詳細に説明する。

【0019】

XOR回路12は、初期処理回路3からのデータS3bとF関数回路11からのデータS11との排他的論理和を演算し、その結果であるデータS12を後段のFeistel構造モジュール4_2のF関数回路11に出力する。

また、F関数回路11は、初期処理回路3からのデータS3aを後段のFeistel構造モジュール4_2のXOR回路12に出力するように構成されている。

その前段のFeistel構造モジュール4_N-1からのデータS3aを下位64ビットとし、Feistel構造モジュール4_NのXOR回路12からのデータS12を上位64ビットとした128ビットのデータが、最終段のFeistel構造モジュール4_Nから後処理回路5に出力される。

10

20

30

40

50

【0020】

後処理回路5は、Feistel構造モジュール4__Nからの128ビットのデータに対して、鍵生成回路2からの鍵データK3を用いて後処理を行い、その結果である暗号化データCTを出力する。

【0021】

以下、図1に示すF関数回路11の構成について説明する。

図2は、図1に示すF関数回路11の構成図である。

図2に示すように、F関数回路11は、例えば、XOR部21、非線形変換部22、線形変換部23、XOR部24および非線形変換部25を有する。

ここで、非線形変換部22が第5の発明の非線形変換回路に対応している。

10

また、線形変換部23が第1の発明～第3の発明を用いた設計対象となり、第4および第5の発明の線形変換回路に対応している。

【0022】

F関数回路11では、入力された64ビットのデータS3aが、各々8ビットの8個のデータモジュールに分割されて処理される。

XOR部21は、データS3aを分割して得られた8個のデータモジュールの各々に対して、鍵生成回路2から入力した鍵データK1との排他的論理和演算を施し、その結果をそれぞれ非線形変換部22に出力する。

【0023】

非線形変換部22は、上記8個のデータモジュールに対応してそれぞれ設けられた非線形変換回路31を有し、非線形変換回路31において、入力したデータモジュールに対して非線形変換処理を施し、その結果である8個のデータモジュールを線形変換部23に出力する。

20

非線形変換回路31は、例えば、S-boxと呼ばれる。

【0024】

線形変換部23は、非線形変換部22からのデータをバイト単位でGF(2)上の演算である拡散処理を行う。

線形変換部23は、非線形変換部22から入力した8個のデータモジュールをそれぞれ伝送する8個のデータ線26__1～26__8（第4および第5の発明のデータ線）を有する。

30

線形変換部23は、図2に示すように、直列に接続された4個の回路ブロック41__1～41__4（第4および第5の発明の回路ブロック）を有する。

回路ブロック41__1は、データ線26__5～26__8上の各々にXOR回路（第4および第5の発明の演算回路）を配設している。

また、データ線26__1、26__2、26__3、26__4上のデータモジュールが、各々データ線26__5、26__6、26__7、26__8上のXOR回路に入力されるように配線されている。

回路ブロック41__2は、データ線26__1～26__4上の各々にXOR回路を配設している。

【0025】

40

また、データ線26__5、26__6、26__7、26__8上のデータモジュールが、各々データ線26__3、26__4、26__1、26__2上のXOR回路に入力されるように配線されている。

また、データ線26__1、26__2、26__3、26__4上のデータモジュールが、各々データ線26__6、26__7、26__8、26__5上のXOR回路に入力されるように配線されている。

また、データ線26__5、26__6、26__7、26__8上のデータモジュールが、各々データ線26__1、26__2、26__3、26__4上のXOR回路に入力されるように配線されている。

【0026】

50

X O R 部 2 4 は、線形変換部 2 3 から入力した 8 個のデータモジュールの各々に対して、鍵生成回路 2 から入力した鍵データ K 3 との排他的論理和演算を施し、その結果をそれぞれ非線形変換部 2 5 に出力する。

【0027】

非線形変換部 2 5 は、X O R 部 2 4 からの上記 8 個のデータモジュールに対応してそれぞれ設けられた非線形変換回路 3 2 を有し、非線形変換回路 3 2 において、入力したデータモジュールに対して非線形変換処理を施し、その結果である 8 個のデータモジュールを出力する。

非線形変換回路 3 2 は、例えば、S - b o x と呼ばれる。

非線形変換部 2 5 から出力された 8 個のデータモジュールは、図 2 に示すように、組み合わされて図 1 に示すデータ S 1 1 として X O R 回路 1 2 に出力される。 10

【0028】

以下、図 2 に示す F 関数回路 1 1 の線形変換部 2 3 の設計方法について説明する。

【0029】

図 3 は、図 2 に示す F 関数回路 1 1 の線形変換部 2 3 の設計に用いられるコンピュータ 3 9 を説明するための図である。

図 3 に示すように、コンピュータ 3 9 は、例えば、メモリ 5 1、操作部 5 2、ディスプレイ 5 3 および C P U 5 4 を有し、これらがバス 5 0 を介して接続されている。

ここで、コンピュータ 3 9 が第 3 の発明のデータ処理装置に対応している。

メモリ 5 1 は、コンピュータ 3 9 が実行するプログラム 4 8 (第 2 の発明のプログラム) 、並びにコンピュータ 3 9 によるプログラム 4 8 の実行に用いられる種々のデータを記憶する。 20

操作部 5 2 は、キーボードやマウス等であり、ユーザによる操作に応じた操作信号を C P U 5 4 に出力する。

ディスプレイ 5 3 は、コンピュータ 3 9 の処理結果を表示する。

C P U 5 4 は、メモリ 5 1 から読み出したプログラム 5 8 を実行し、図 2 に示す F 関数回路 1 1 の設計処理を行う。

C P U 5 4 は、複数の線形変換候補のうち、当該線形変換候補を実現する回路上の制約を満たす複数の前記線形変換候補を特定し、当該特定した線形変換候補のそれぞれについて複数の入力データを基に線形変換処理を行い、それらの処理結果内に生じる零の数の最小値 (いわゆるアクティブ S - b o x) を求め、その最小値を最大とする線形変換候補を特定する。そして、C P U 5 4 は、当該特定した線形変換候補を基に、図 2 に示す線形変換部 2 3 を構成する。 30

【0030】

以下、C P U 5 4 の設計処理手順 (本実施形態の設計方法) を説明する。

図 4 は、C P U 5 4 の設計処理手順を説明するためのフローチャートである。

以下の手順の一部は、例えば、C P U 5 4 がディスプレイ 5 3 に表示した操作画面を基にユーザが行った操作部 5 2 の操作に応じて、ユーザと C P U 5 4 との間で対話形式で行われる。

なお、図 4 に示すステップ S T 1 ~ S T 3 が第 1 の発明の第 1 の工程、第 2 の発明の第 1 の手順、並びに第 3 の発明の第 1 の手段に対応している。 40

ステップ S T 4 ~ S T 9 が第 1 の発明の第 2 の工程、第 2 の発明の第 2 の手順、並びに第 3 の発明の第 2 の手段に対応している。

ステップ S T 1 0 が第 1 の発明の第 3 の工程、第 2 の発明の第 3 の手順、並びに第 3 の発明の第 3 の手段に対応している。

【0031】

ステップ S T 1 :

ユーザは、設計対象の図 2 に示す線形変換部 2 3 の線形変換を例えば、4 つの線形変換ブロックに分割して規定し、その情報を操作部 5 2 を介して C P U 5 4 に与える。

【0032】

ステップ S T 2 :

C P U 5 4 は、ステップ S T 1 で受けた情報を基に、下記式 (2) , (3) , (4) , (5) に示すように、各々 4×4 の行列である変数行列 C_1 , C_2 , C_3 , C_4 を用いて、行列 D_1 , D_2 , D_3 , D_4 を規定する。

このように、各々 8×8 の行列である行列 D_1 , D_2 , D_3 , D_4 を規定することで、行列 D_1 , D_2 , D_3 , D_4 を実現した図 2 に示す回路ブロック 4 1 _ 1 ~ 4 1 _ 4 を、X O R 回路が設けられていないデータ線から、最大 1 個の X O R 回路にデータモジュールを出力するように構成できる。これにより、回路ブロック 4 1 _ 1 ~ 4 1 _ 4 に規定された回路上の制約が満たされる。

具体的には、図 2 に示すように、上位あるいは下位の 4 本のデータ線上に X O R 回路を設け、当該 X O R 回路が設けられていない 4 本のデータ線から、それぞれ異なる上記 X O R 回路にデータモジュールを出力するように回路ブロック 4 1 _ 1 ~ 4 1 _ 4 を構成できる。

すなわち、本実施形態では、上述したように行列 D_1 ~ D_4 を規定することで、回路上の制約を満たさないものについては、行列 D_1 ~ D_4 の候補から予め除外することができる。

【 0 0 3 3 】

【 数 2 】

$$D_1 = \begin{pmatrix} I & O \\ C_1 & I \end{pmatrix} \cdots (2)$$

$$D_2 = \begin{pmatrix} I & C_2 \\ O & I \end{pmatrix} \cdots (3)$$

$$D_3 = \begin{pmatrix} I & O \\ C_3 & I \end{pmatrix} \cdots (4)$$

$$D_4 = \begin{pmatrix} I & C_4 \\ O & I \end{pmatrix} \cdots (5)$$

【 0 0 3 4 】

ステップ S T 3 :

C P U 5 4 は、下記式 (6) , (7) に示すように、ステップ S T 2 で規定した行列 D_1 , D_2 , D_3 , D_4 を合成した行列 A (本発明の線形変換候補) を算出する。

【 0 0 3 5 】

【 数 3 】

$$A = D_1 D_2 D_3 D_4 \cdots (6)$$

【 0 0 3 6 】

【 数 4 】

20

30

40

$$\begin{aligned}
 A &= \begin{pmatrix} I & C_4 \\ O & I \end{pmatrix} \begin{pmatrix} I & O \\ C_3 & I \end{pmatrix} \begin{pmatrix} I & C_2 \\ O & I \end{pmatrix} \begin{pmatrix} I & O \\ C_1 & I \end{pmatrix} \\
 &= \begin{pmatrix} I + C_4 C_3 + C_2 C_1 + C_4 C_3 C_2 C_1 + C_4 C_1 & C_2 + C_4 C_3 C_2 + C_4 \\ C_3 + C_3 C_2 C_1 + C_1 & I + C_3 C_2 \end{pmatrix} \\
 &\quad \dots (7)
 \end{aligned}$$

【0037】

ステップST4:

CPU54は、ステップST3で生成した行列A内の変数行列 C_1 , C_2 , C_3 , C_4 の各要素に所定の値を与える。

これにより、本発明にいう、「複数の線形変換候補の特定」が行われる。

なお、CPU54は、ステップST9からのループバックにより当該ステップST4の処理を複数回行い、その度に、異なる行列Aを規定する。

【0038】

ステップST5:

CPU54は、予め決められた複数の入力データ（本発明の入力データ）のうち、次に行列Aに入力する、すなわち行列Aによる演算対象とする入力データを決定する。

なお、CPU54は、ステップST8からのループバックにより当該ステップST5の処理を複数回行い、その度に、異なる入力データを決定する。

【0039】

ステップST6:

CPU54は、ステップST5で決定した入力データに対してステップST4で決定した行列Aの演算を行う。

ステップST7:

CPU54は、ステップST6の演算結果（64ビットデータ）内に含まれる零（0）の数を計数し、その計数値が、それまで計数した最小値より小さい場合に最小値を更新する。CPU54は、全ての行列Aの各々について当該最小値（本発明の最小値）を求める。

【0040】

ステップST8:

CPU54は、上記予め決められた全ての入力データについて、ステップST6の処理を行ったか否かを判断し、行っていると判断した場合にステップST9の処理に進み、そうでない場合にステップST5の処理に戻る。

ステップST9:

CPU54は、変数行列 C_1 , C_2 , C_3 , C_4 を用いて規定可能な全ての行列Aについて、ステップST6の処理を行ったか否かを判断し、行っていると判断した場合にステップST10の処理に進み、そうでない場合にステップST4の処理に戻る。

【0041】

ステップST10:

CPU54は、ステップST2で最終的に得られた全ての行列Aの最小値のうち、最大の最小値を出した行列Aを特定する。

ステップST11:

CPU54は、ステップST10で特定した行列Aで用いられた変数行列 C_1 , C_2 , C_3 , C_4 を用いて図2示す線形変換部23の回路ブロック41_1～41_4を構成（設計）する。

【0042】

以上説明したように、本実施形態の設計方法によれば、複数の線形変換候補のうち、当該線形変換候補を実現する回路上の制約を満たす複数の前記線形変換候補を特定し、当該特定した線形変換候補のなかから、線形変換を行った結果に零が生じる個数の最小値を最大

10

20

30

40

50

にする線形演算候補を探索（特定）するため、全ての線形変換候補を対象としてを探索を行う場合に比べて、演算量を大幅に削減できる。

具体的には、本実施形態の設計方法によれば、変数行列 $C_1 \sim C_4$ の各々は 4×4 の行列であるため、 $4! (4 \text{ の階乗})$ 通りの候補がある。従って、 (C_1, C_2, C_3, C_4) の組み合わせは、 $(4!)^4$ 通り、すなわち線形変換候補は、約 2^{18} 通りになる。

これに対して、従来では、 $GF(2)$ 上の 8×8 行列の全てを線形変換候補するため、線形変換候補は 2^{64} 通りになる。

従って、本実施形態の設計方法によれば、設計に伴う演算量を従来に比べて大幅に削減できる。これにより、本実施形態によれば、線形変換部 23 の設計を実用的な時間で行うことが可能になる。

10

【0043】

本発明は上述した実施形態には限定されない。

上述した実施形態では、 $GF(2)$ 上の変換をバイト単位で行い、32ビット演算を高速に行う場合を例示したが、本発明は、例えば、16ビットワードを $GF(2)$ 上の変換の単位として64ビット演算を行ったり、バイトを $GF(2)$ 上の変換の単位として64ビット演算を行うことで、上記線形変換を行うように設計を行ってもよい。

【0044】

【発明の効果】

以上説明したように、本発明によれば、複数の線形変換候補のなかから、複数の入力データに線形変換を行った結果に零が生じる個数の最小値が最大となる線形変換候補を従来に比べて少ない演算量で特定できるデータ処理方法、その装置および、そのプログラムを提供できる。

20

また、本発明によれば、上述した本発明のデータ処理方法、その装置およびそのプログラムによって設計される線形変換回路および暗号化装置を提供できる。

【図面の簡単な説明】

【図1】図1は、本発明の実施形態に係わる暗号化装置の構成図である。

【図2】図2は、図1に示すF関数回路の構成図である。

【図3】図3は、図2に示すF関数回路の線形変換部の設計に用いられるコンピュータを説明するための図である。

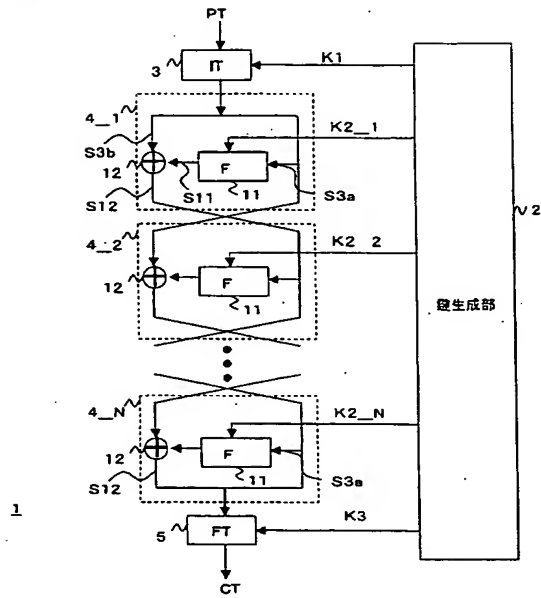
30

【図4】図4は、図3に示すCPUの設計処理手順を説明するためのフローチャートである。

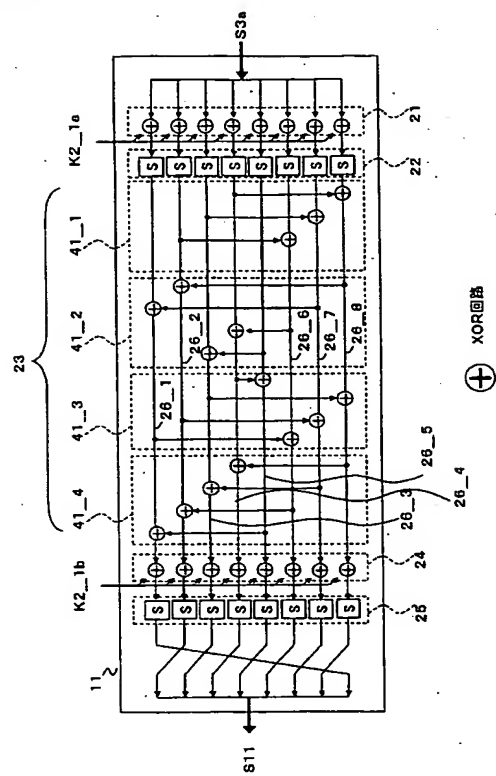
【符号の説明】

1…暗号化装置、2…鍵生成回路、3…初期処理回路、4_1～4_N…Feistel構造モジュール、5…後処理回路、11…F関数回路、12…XOR回路、21…XOR部、22…非線形変換部、23…線形変換部、24…XOR部、25…非線形変換部、41_1～41_4…回路ブロック、39…コンピュータ、51…メモリ、52…操作部、53…ディスプレイ、54…CPU、58…プログラム

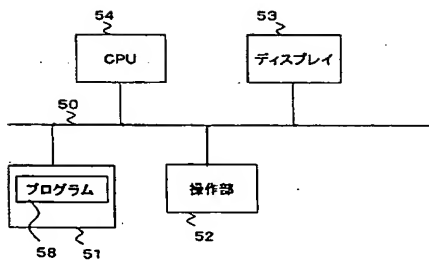
【図 1】



【図 2】



【図 3】



【図 4】

